# Dependency Pairs Revisited

Nao Hirokawa and Aart Middeldorp

Institute of Computer Science
University of Innsbruck
6020 Innsbruck, Austria

**Abstract.** In this paper we present some new refinements of the dependency pair method for automatically proving the termination of term rewrite systems. These refinements are very easy to implement, increase the power of the method, result in simpler termination proofs, and make the method more efficient.

## 1 Introduction

Since the introduction of the dependency pair method (Arts and Giesl [1]) and the monotonic semantic path order (Borralleras, Ferreira, and Rubio [5]), two powerful methods that facilitate termination proofs that can be obtained automatically, there is a renewed interest in the study of termination for term rewrite systems. Three important issues which receive a lot of attention in current research on termination are to make these methods faster, to improve the methods such that more and more (challenging) rewrite systems can be handled, and to extend the methods beyond the realm of ordinary first-order term rewriting. Especially in connection with the dependency pair method many improvements, extensions, and refinements have been published. The method forms an important ingredient in several software tools for proving terminating. To mention a few (in order of appearance): CiME [6], T$\top$T [15], AProVE [12], and TORPA [24].

In this paper we go back to the foundations of the dependency pair method. Starting from scratch, we give a systematic account of the method. Along the way we derive two new refinements which are very easy to implement, increase the termination proving power,[1] give rise to simpler termination proofs, and make the method much faster.

We use the following term rewrite system (TRS for short) from Dershowitz [7] to illustrate the developments in the remainder of the paper:

$$\begin{aligned}
1&: & \neg\neg x &\to x\\
2&: & \neg(x \vee y) &\to \neg x \wedge \neg y\\
3&: & \neg(x \wedge y) &\to \neg x \vee \neg y\\
4&: & x \wedge (y \vee z) &\to (x \wedge y) \vee (x \wedge z)\\
5&: & (y \vee z) \wedge x &\to (x \wedge y) \vee (x \wedge z)
\end{aligned}$$

[1] Note however the discussion at the end of Section 5.

Termination of this TRS is easily shown by the multiset path order. This, however, does not mean that automatic termination tools easily find a termination proof. For instance, both CiME and the fully automatic "Meta Combination" algorithm in AProVE 1.0 fail to prove termination.

We assume familiarity with the basics of term rewriting ([3, 20]). We just recall some basic notation and terminology. The set of terms $\mathcal{T}(\mathcal{F}, \mathcal{V})$ constructed from a signature $\mathcal{F}$ and a disjoint set $\mathcal{V}$ of variables is abbreviated to $\mathcal{T}$ when no confusion can arise. The set of variables appearing in a term $t$ is denoted by $\mathcal{V}\mathsf{ar}(t)$. The root symbol of a term $t$ is denoted by $\mathrm{root}(t)$. Defined function symbols are root symbols of left-hand sides of rewrite rules. We use $\xrightarrow{\epsilon}$ to denote root rewrite steps and $\xrightarrow{>\epsilon}$ to denote rewrite steps in which the selected redex occurs below the root. A substitution is a mapping $\sigma$ from variables to terms such that its domain $\mathcal{D}\mathsf{om}(\sigma) = \{x \mid x \neq \sigma(x)\}$ is finite. We write $t\sigma$ to denote the result of applying the substitution $\sigma$ to the term $t$. A relation $R$ on terms is closed under substitutions if $(s\sigma, t\sigma) \in R$ whenever $(s, t) \in R$, for all substitutions $\sigma$. We say that $R$ is closed under contexts if $(u[s]_p, u[t]_p) \in R$ whenever $(s, t) \in R$, for all terms $u$ and positions $p$ in $u$. The superterm relation is denoted by $\trianglerighteq$ (i.e., $s \trianglerighteq t$ if $t$ is a subterm of $s$) and $\triangleright$ denotes its strict part.

## 2 Dependency Pairs

In this section and in Section 4 we recall the basics of the dependency pair method of Arts and Giesl [1]. We provide proofs of all results.

Let us start with some easy observations. If a TRS $\mathcal{R}$ is not terminating then there must be a *minimal* non-terminating term, minimal in the sense that all its proper subterms are terminating. Let us denote the set of all minimal non-terminating terms by $\mathcal{T}_\infty$.

**Lemma 1.** *For every term $t \in \mathcal{T}_\infty$ there exists a rewrite rule $l \to r$, a substitution $\sigma$, and a non-variable subterm $u$ of $r$ such that $t \xrightarrow{>\epsilon}^* l\sigma \xrightarrow{\epsilon} r\sigma \trianglerighteq u\sigma$ and $u\sigma \in \mathcal{T}_\infty$.*

*Proof.* Let $A$ be an infinite rewrite sequence starting at $t$. Since all proper subterms of $t$ are terminating, $A$ must contain a root rewrite step. By considering the first root rewrite step in $A$ it follows that there exist a rewrite rule $l \to r$ and a substitution $\sigma$ such that $A$ starts with $t \xrightarrow{>\epsilon}^* l\sigma \xrightarrow{\epsilon} r\sigma$. Write $l = f(l_1, \dots, l_n)$. Since the rewrite steps in $t \to^* l\sigma$ take place below the root, $t = f(t_1, \dots, t_n)$ and $t_i \to^* l_i\sigma$ for all $1 \leqslant i \leqslant n$. By assumption the arguments $t_1, \dots, t_n$ of $t$ are terminating. Hence so are the terms $l_1\sigma, \dots, l_n\sigma$. It follows that $\sigma(x)$ is terminating for every $x \in \mathcal{V}\mathsf{ar}(r) \subseteq \mathcal{V}\mathsf{ar}(l)$. As $r\sigma$ is non-terminating it has a subterm $t' \in \mathcal{T}_\infty$. Because non-terminating terms cannot occur in the substitution part, there must be a non-variable subterm $u$ of $r$ such that $t' = u\sigma$. □

Observe that the term $l\sigma$ in Lemma 1 belongs to $\mathcal{T}_\infty$ as well. Further note that $u\sigma$ cannot be a proper subterm of $l\sigma$ (since all arguments of $l\sigma$ are terminating).

**Corollary 2.** *Every term in $\mathcal{T}_\infty$ has a defined root symbol.* □

If we were to define a new TRS $\mathcal{S}$ consisting of all rewrite rules $l \to u$ for which there exist a rewrite rule $l \to r \in \mathcal{R}$ and a subterm $u$ of $r$ with defined function symbol, then the sequence in the conclusion of Lemma 1 is of the form $\xrightarrow{>\epsilon}{}^{*}_{\mathcal{R}} \cdot \xrightarrow{\epsilon}_{\mathcal{S}}$. The idea is now to get rid of the position constraints by marking the root symbols of the terms in the rewrite rules of $\mathcal{S}$.

**Definition 3.** *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$. Let $\mathcal{F}^\sharp$ denote the union of $\mathcal{F}$ and $\{f^\sharp \mid f$ is a defined symbol of $\mathcal{R}\}$ where $f^\sharp$ is a fresh function symbol with the same arity as $f$. We call these new symbols* dependency pair symbols. *Given a term $t = f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ with $f$ a defined symbol, we write $t^\sharp$ for the term $f^\sharp(t_1, \ldots, t_n)$. If $l \to r \in \mathcal{R}$ and $u$ is a subterm of $r$ with defined root symbol such that $u$ is not a proper subterm of $l$ then the rewrite rule $l^\sharp \to u^\sharp$ is called a* dependency pair *of $\mathcal{R}$. The set of all dependency pairs of $\mathcal{R}$ is denoted by $\mathsf{DP}(\mathcal{R})$.*

The idea of excluding dependency pairs $l^\sharp \to u^\sharp$ where $u$ is a proper subterm of $l$ is due to Dershowitz [8]. Although dependency pair symbols are defined symbols of $\mathsf{DP}(\mathcal{R})$, they are not defined symbols of $\mathcal{R}$. In the following, defined symbols always refer to the original TRS $\mathcal{R}$.

*Example 4.* The example in the introduction admits the following 9 dependency pairs:

$$\begin{array}{rlrl}
6: & \neg^\sharp(x \vee y) \to \neg x \wedge^\sharp \neg y & & \\
7: & \neg^\sharp(x \vee y) \to \neg^\sharp x & 11: & x \wedge^\sharp (y \vee z) \to x \wedge^\sharp y \\
8: & \neg^\sharp(x \vee y) \to \neg^\sharp y & 12: & x \wedge^\sharp (y \vee z) \to x \wedge^\sharp z \\
9: & \neg^\sharp(x \wedge y) \to \neg^\sharp x & 13: & (y \vee z) \wedge^\sharp x \to x \wedge^\sharp y \\
10: & \neg^\sharp(x \wedge y) \to \neg^\sharp y & 14: & (y \vee z) \wedge^\sharp x \to x \wedge^\sharp z
\end{array}$$

**Lemma 5.** *For every term $s \in \mathcal{T}_\infty$ there exist terms $t, u \in \mathcal{T}_\infty$ such that $s^\sharp \to^*_{\mathcal{R}} t^\sharp \to_{\mathsf{DP}(\mathcal{R})} u^\sharp$.*

*Proof.* Immediate from Lemma 1, Corollary 2, and the preceding definition. □

**Definition 6.** *For any subset $T \subseteq \mathcal{T}$ consisting of terms with a defined root symbol, we denote the set $\{t^\sharp \mid t \in T\}$ by $T^\sharp$.*

An immediate consequence of the previous lemma is that for every non-terminating TRS $\mathcal{R}$ there exists an infinite rewrite sequence of the form

$$t_1 \to^*_{\mathcal{R}} t_2 \to_{\mathsf{DP}(\mathcal{R})} t_3 \to^*_{\mathcal{R}} t_4 \to_{\mathsf{DP}(\mathcal{R})} \cdots$$

with $t_i \in \mathcal{T}^\sharp_\infty$ for all $i \geqslant 1$. Hence, to prove termination of a TRS $\mathcal{R}$ it is sufficient to show that $\mathcal{R} \cup \mathsf{DP}(\mathcal{R})$ does not admit such infinite sequences. Every such sequence contains a tail in which all applied dependency pairs are used infinitely many times. For finite TRSs, the set of those dependency pairs forms a cycle in the dependency graph. *From now on, we assume that all TRSs are finite.*

As a side remark, note that all terms in $\mathcal{T}^\sharp_\infty$ are terminating with respect to $\mathcal{R}$ but admit an infinite rewrite sequence with respect to $\mathcal{R} \cup \mathsf{DP}(\mathcal{R})$.

**Definition 7.** *The nodes of the* dependency graph $\mathsf{DG}(\mathcal{R})$ *are the dependency pairs of* $\mathcal{R}$ *and there is an arrow from* $s \to t$ *to* $u \to v$ *if and only if there exist substitutions* $\sigma$ *and* $\tau$ *such that* $t\sigma \to^*_{\mathcal{R}} u\tau$. *A* cycle *is a nonempty subset* $\mathcal{C}$ *of dependency pairs of* $\mathsf{DP}(\mathcal{R})$ *if for every two (not necessarily distinct) pairs* $s \to t$ *and* $u \to v$ *in* $\mathcal{C}$ *there exists a nonempty path in* $\mathcal{C}$ *from* $s \to t$ *to* $u \to v$.

**Definition 8.** *Let* $\mathcal{C} \subseteq \mathsf{DP}(\mathcal{R})$. *An infinite rewrite sequence in* $\mathcal{R} \cup \mathcal{C}$ *of the form*

$$t_1 \to^*_{\mathcal{R}} t_2 \to_{\mathcal{C}} t_3 \to^*_{\mathcal{R}} t_4 \to_{\mathcal{C}} \cdots$$

*with* $t_1 \in \mathcal{T}^\sharp_\infty$ *is called* $\mathcal{C}$-minimal *if all rules in* $\mathcal{C}$ *are applied infinitely often.*

Hence proving termination boils down to proving the absence of $\mathcal{C}$-minimal rewrite sequences, for any cycle $\mathcal{C}$ in the dependency graph $\mathsf{DG}(\mathcal{R})$.

*Example 9.* The example in the introduction has the following dependency graph:

$$\begin{tikzcd}
7 \arrow[loop left] \arrow[r, leftrightarrow] \arrow[dd, leftrightarrow] \arrow[ddr, leftrightarrow] \arrow[drr, bend left] & 8 \arrow[loop right] \arrow[dd, leftrightarrow] \arrow[ddl, leftrightarrow] \arrow[dr] & & 11 \arrow[loop left] \arrow[r, leftrightarrow] \arrow[dd, leftrightarrow] \arrow[ddr, leftrightarrow] & 12 \arrow[loop right] \arrow[dd, leftrightarrow] \arrow[ddl, leftrightarrow] \\
& & 6 \arrow[ur] \arrow[dr] \arrow[urr, bend left] \arrow[drr, bend right] & & \\
9 \arrow[loop left] \arrow[r, leftrightarrow] \arrow[urr, bend right] & 10 \arrow[loop right] \arrow[ur] & & 13 \arrow[loop left] \arrow[r, leftrightarrow] & 14 \arrow[loop right]
\end{tikzcd}$$

It contains 30 cycles: all nonempty subsets of both $\{7, 8, 9, 10\}$ and $\{11, 12, 13, 14\}$.

Although the dependency graph is not computable in general, sound approximations exist that can be computed efficiently (see [1, 17]). Soundness here means that every cycle in the real dependency graph is a cycle in the approximated graph. For the example TRS all known approximations compute the real dependency graph.

## 3 Subterm Criterion

We now present a new criterion which permits us to ignore certain cycles of the dependency graph.

**Definition 10.** *Let $\mathcal{R}$ be a TRS and $\mathcal{C} \subseteq \mathsf{DP}(\mathcal{R})$ such that every dependency pair symbol in $\mathcal{C}$ has positive arity. A* simple projection *for $\mathcal{C}$ is a mapping $\pi$ that assigns to every $n$-ary dependency pair symbol $f^\sharp$ in $\mathcal{C}$ an argument position $i \in \{1, \ldots, n\}$. The mapping that assigns to every term $f^\sharp(t_1, \ldots, t_n) \in \mathcal{T}^\sharp$ with $f^\sharp$ a dependency pair symbol in $\mathcal{C}$ its argument at position $\pi(f^\sharp)$ is also denoted by $\pi$.*

**Theorem 11.** *Let $\mathcal{R}$ be a TRS and let $\mathcal{C}$ be a cycle in $\mathsf{DG}(\mathcal{R})$. If there exists a simple projection $\pi$ for $\mathcal{C}$ such that $\pi(\mathcal{C}) \subseteq \trianglerighteq$ and $\pi(\mathcal{C}) \cap \vartriangleright \neq \varnothing$ then there are no $\mathcal{C}$-minimal rewrite sequences.*

Before presenting the proof, let us make some clarifying remarks about the notation. If $R$ is a set of rewrite rules and $O$ is a relation on terms then the expression $\pi(R)$ denotes the set $\{\pi(l) \to \pi(r) \mid l \to r \in R\}$, the inclusion $R \subseteq O$ abbreviates "$(l, r) \in O$ for all $l \to r \in O$", and the inequality $R \cap O \neq \varnothing$ abbreviates "$(l, r) \in O$ for at least one $l \to r \in O$". So the conditions state that after applying the simple projection $\pi$, every rule in $\mathcal{C}$ is turned into an identity or a rule whose right-hand side is a proper subterm of the left-hand side. Moreover, the latter case applies at least once.

*Proof.* Suppose to the contrary that there exists a $\mathcal{C}$-minimal rewrite sequence:

$$t_1 \to_{\mathcal{R}}^* u_1 \to_{\mathcal{C}} t_2 \to_{\mathcal{R}}^* u_2 \to_{\mathcal{C}} t_3 \to_{\mathcal{R}}^* \cdots \tag{1}$$

All terms in this sequence have a dependency pair symbol in $\mathcal{C}$ as root symbol. We apply the simple projection $\pi$ to (1). Let $i \geqslant 1$.

- First consider the dependency pair step $u_i \to_{\mathcal{C}} t_{i+1}$. There exist a dependency pair $l \to r \in \mathcal{C}$ and a substitution $\sigma$ such that $u_i = l\sigma$ and $t_{i+1} = r\sigma$. We have $\pi(u_i) = \pi(l)\sigma$ and $\pi(t_{i+1}) = \pi(r)\sigma$. We have $\pi(l) \trianglerighteq \pi(r)$ by assumption. So $\pi(l) = \pi(r)$ or $\pi(l) \vartriangleright \pi(r)$. In the former case we trivially have $\pi(u_i) = \pi(t_{i+1})$. In the latter case the closure under substitutions of $\vartriangleright$ yields $\pi(u_i) \vartriangleright \pi(t_{i+1})$. Because of the assumption $\pi(\mathcal{C}) \cap \vartriangleright \neq \varnothing$, the latter holds for infinitely many $i$.
- Next consider the rewrite sequence $t_i \to_{\mathcal{R}}^* u_i$. All steps in this sequence take place below the root and thus we obtain the (possibly shorter) sequence $\pi(t_i) \to_{\mathcal{R}}^* \pi(u_i)$.

So by applying the simple projection $\pi$, sequence (1) is transformed into an infinite $\to_{\mathcal{R}} \cup \vartriangleright$ sequence containing infinitely many $\vartriangleright$ steps, starting from the term $\pi(t_1)$. Since the relation $\vartriangleright$ is well-founded, the infinite sequence must also contain infinitely many $\to_{\mathcal{R}}$ steps. By making repeated use of the well-known relational inclusion $\vartriangleright \cdot \to_{\mathcal{R}} \subseteq \to_{\mathcal{R}} \cdot \vartriangleright$ ($\vartriangleright$ commutes over $\to_{\mathcal{R}}$ in the terminology of [4]), we obtain an infinite $\to_{\mathcal{R}}$ sequence starting from $\pi(t_1)$. In other words, the term $\pi(t_1)$ is non-terminating with respect to $\mathcal{R}$. Let $t_1 = f^\sharp(s_1, \ldots, s_n)$. Because $t_1 \in \mathcal{T}_\infty^\sharp$, $f(s_1, \ldots, s_n)$ is a minimal non-terminating term. Consequently, its argument $\pi(t_1) = s_{\pi(f^\sharp)}$ is terminating with respect to $\mathcal{R}$, providing the desired contradiction. ◻

The remarkable thing about the above theorem is that it permits us to discard cycles of the dependency graph without considering any rewrite rules. This is extremely useful. Moreover, the criterion is very simple to check.

*Example 12.* Consider the cycle $\mathcal{C} = \{7, 8, 9, 10\}$. The only dependency pair symbol in $\mathcal{C}$ is $\neg^\sharp$. Since $\neg^\sharp$ is a unary function symbol, there is just one simple projection for $\mathcal{C}$: $\pi(\neg^\sharp) = 1$. By applying $\pi$ to $\mathcal{C}$, we obtain

$$\begin{aligned} 7: &\quad x \vee y \to x \\ 8: &\quad x \vee y \to y \\ 9: &\quad x \wedge y \to x \\ 10: &\quad x \wedge y \to y \end{aligned}$$

We clearly have $\pi(\mathcal{C}) \subseteq \rhd$. Hence we can ignore $\mathcal{C}$ (and all its subcycles). The only cycles that are not handled by the criterion of Theorem 11 are the ones that involve 13 or 14; applying the simple projection $\pi(\wedge^\sharp) = 1$ produces

$$\begin{aligned} 13: &\quad y \vee z \to x \\ 14: &\quad y \vee z \to x \end{aligned}$$

whereas $\pi(\wedge^\sharp) = 2$ gives

$$\begin{aligned} 13: &\quad x \to y \\ 14: &\quad x \to z \end{aligned}$$

None of these rules are compatible with $\trianglerighteq$.

In implementations one shouldn't compute all cycles of the dependency graph (since there can be exponentially many in the number of dependency pairs), but use the technique of Hirokawa and Middeldorp [14] to recursively solve strongly connected components (which gives rise to a linear algorithm): if all pairs in a strongly connected component (SCC for short) are compatible with $\trianglerighteq$ after applying a simple projection, the ones that are compatible with $\rhd$ are removed and new SCCs among the remaining pairs are computed. This is illustrated in the final two examples in this section. The last example furthermore shows that the subterm criterion is capable of proving the termination of TRSs that are considered to be challenging in the termination literature (cf. the remarks in [10, Example 9]).

*Example 13.* Consider the following TRS from [7]:

$$\begin{aligned} 1: &\quad & \mathsf{sort}([\,]) &\to [\,] \\ 2: &\quad & \mathsf{sort}(x : y) &\to \mathsf{insert}(x, \mathsf{sort}(y)) \\ 3: &\quad & \mathsf{insert}(x, [\,]) &\to x : [\,] \\ 4: &\quad & \mathsf{insert}(x, v : w) &\to \mathsf{choose}(x, v : w, x, v) \\ 5: &\quad & \mathsf{choose}(x, v : w, y, \mathsf{0}) &\to x : (v : w) \\ 6: &\quad & \mathsf{choose}(x, v : w, \mathsf{0}, \mathsf{s}(z)) &\to v : \mathsf{insert}(x, w) \\ 7: &\quad & \mathsf{choose}(x, v : w, \mathsf{s}(y), \mathsf{s}(z)) &\to \mathsf{choose}(x, v : w, y, z) \end{aligned}$$

There are 5 dependency pairs:

$$\begin{aligned}
8\colon&& \mathsf{sort}^\sharp(x:y) &\to \mathsf{insert}^\sharp(x,\mathsf{sort}(y))\\
9\colon&& \mathsf{sort}^\sharp(x:y) &\to \mathsf{sort}^\sharp(y)\\
10\colon&& \mathsf{insert}^\sharp(x,v:w) &\to \mathsf{choose}^\sharp(x,v:w,x,v)\\
11\colon&& \mathsf{choose}^\sharp(x,v:w,\mathsf{0},\mathsf{s}(z)) &\to \mathsf{insert}^\sharp(x,w)\\
12\colon&& \mathsf{choose}^\sharp(x,v:w,\mathsf{s}(y),\mathsf{s}(z)) &\to \mathsf{choose}^\sharp(x,v:w,y,z)
\end{aligned}$$

The dependency graph

$$\begin{tikzcd}
9 \arrow[loop left] \arrow[r] & 8 \arrow[r] & 10 \arrow[r, leftrightarrow] \arrow[rr, bend left] & 11 & 12 \arrow[l] \arrow[loop right]
\end{tikzcd}$$

contains 2 SCCs: $\{9\}$ and $\{10,11,12\}$. The first one is handled by the simple projection $\pi(\mathsf{sort}^\sharp)=1$:

$$9\colon\quad x:y\to x$$

For the other SCC we take $\pi(\mathsf{insert}^\sharp)=\pi(\mathsf{choose}^\sharp)=2$:

$$\begin{aligned}
10\colon&\quad v:w\to v:w\\
11\colon&\quad v:w\to w\\
12\colon&\quad v:w\to v:w
\end{aligned}$$

After removing the strictly decreasing pair 11, we are left with 10 and 12. The restriction of the dependency graph to these two pairs contains one SCC: $\{12\}$. This pair is handled by the simple projection $\pi(\mathsf{choose}^\sharp)=3$:

$$12\colon\quad s(y)\to y$$

Hence the TRS is terminating.

*Example 14.* Consider the following TRS from [19]:

$$\begin{aligned}
1\colon&& \mathsf{intlist}([\,]) &\to [\,]\\
2\colon&& \mathsf{intlist}(x:y) &\to \mathsf{s}(x):\mathsf{intlist}(y)\\
3\colon&& \mathsf{int}(\mathsf{0},\mathsf{0}) &\to \mathsf{0}:[\,]\\
4\colon&& \mathsf{int}(\mathsf{0},\mathsf{s}(y)) &\to \mathsf{0}:\mathsf{int}(\mathsf{s}(\mathsf{0}),\mathsf{s}(y))\\
5\colon&& \mathsf{int}(\mathsf{s}(x),\mathsf{0}) &\to [\,]\\
6\colon&& \mathsf{int}(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{intlist}(\mathsf{int}(x,y))
\end{aligned}$$

There are 4 dependency pairs:

$$\begin{aligned}
7\colon&& \mathsf{intlist}^\sharp(x:y) &\to \mathsf{intlist}^\sharp(y)\\
8\colon&& \mathsf{int}^\sharp(\mathsf{0},\mathsf{s}(y)) &\to \mathsf{int}^\sharp(\mathsf{s}(\mathsf{0}),\mathsf{s}(y))\\
9\colon&& \mathsf{int}^\sharp(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{intlist}^\sharp(\mathsf{int}(x,y))\\
10\colon&& \mathsf{int}^\sharp(\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{int}^\sharp(x,y)
\end{aligned}$$

The dependency graph

$$\begin{tikzcd}
8 \arrow[r, leftrightarrow] \arrow[rr, bend left] & 10 \arrow[r] \arrow[loop below] & 9 \arrow[r] & 7 \arrow[loop below]
\end{tikzcd}$$

contains 2 SCCs: $\{7\}$ and $\{8, 10\}$. The first one is handled by the simple projection $\pi(\mathsf{intlist}^\sharp) = 1$:

$$7\colon \quad x : y \to y$$

For the second one we use the simple projection $\pi(\mathsf{int}^\sharp) = 2$:

$$\begin{aligned} 8\colon &\quad \mathsf{s}(y) \to \mathsf{s}(y) \\ 10\colon &\quad \mathsf{s}(y) \to y \end{aligned}$$

After removing the strictly decreasing pair 10, we are left with 8. Since the restriction of the dependency graph to the remaining pair 8 contains no SCCs, the TRS is terminating.

An empirical evaluation of the subterm criterion can be found in Section 6.

## 4 Reduction Pairs and Argument Filterings

What to do with cycles $\mathcal{C}$ of the dependency graph that cannot be handled by the criterion of the preceding section? In the dependency pair approach one uses a pair of orderings $(\gtrsim, >)$ that satisfy the properties stated below such that (1) all rules in $\mathcal{R}$ are oriented by $\gtrsim$, (2) all rules in $\mathcal{C}$ are oriented by $\gtrsim \cup >$, and (3) at least one rule in $\mathcal{C}$ is oriented by $>$.

**Definition 15.** *A* rewrite preorder *is a preorder (i.e., a transitive and reflexive relation) on terms which is closed under contexts and substitutions. A* reduction pair $(\gtrsim, >)$ *consists of a rewrite preorder* $\gtrsim$ *and a compatible well-founded order* $>$ *which is closed under substitutions. Compatibility means that the inclusion* $\gtrsim \cdot > \subseteq >$ *or the inclusion* $> \cdot \gtrsim \subseteq >$ *holds.*

Since we do not demand that $>$ is the strict part of the preorder $\gtrsim$, the identity $\gtrsim \cdot > = >$ need not hold, although the reduction pairs that are used in practice do satisfy this identity.

A typical example of a reduction pair is $(\geq_{\text{lpo}}, >_{\text{lpo}})$, where $>_{\text{lpo}}$ is the lexicographic path order induced by the (strict) precedence $>$ and $\geq_{\text{lpo}}$ denotes its reflexive closure. Both $\geq_{\text{lpo}}$ and $>_{\text{lpo}}$ are closed under contexts and the identity $\geq_{\text{lpo}} \cdot >_{\text{lpo}} = >_{\text{lpo}}$ holds.

A general semantic construction of reduction pairs, which covers polynomial interpretations, is based on the concept of algebra. If we equip the carrier $A$ of an algebra $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$ with a well-founded order $>$ such that every interpretation function is weakly monotone in all arguments (i.e.,

$f_\mathcal{A}(x_1,\dots,x_n) \geqslant f_\mathcal{A}(y_1,\dots,y_n)$ whenever $x_i \geqslant y_i$ for all $1 \leqslant i \leqslant n$, for every $n$-ary function symbol $f \in \mathcal{F}$) then $(\gtrsim_\mathcal{A}, >_\mathcal{A})$ is a reduction pair. Here the relations $\gtrsim_\mathcal{A}$ and $>_\mathcal{A}$ are defined as follows: $s \gtrsim_\mathcal{A} t$ if $[\alpha]_\mathcal{A}(s) \geqslant [\alpha]_\mathcal{A}(t)$ and $s >_\mathcal{A} t$ if $[\alpha]_\mathcal{A}(s) > [\alpha]_\mathcal{A}(t)$, for all assignments $\alpha$ of elements of $A$ to the variables in $s$ and $t$ ($[\alpha]_\mathcal{A}(\cdot)$ denotes the usual evaluation function associated with the algebra $\mathcal{A}$). In general, the relation $>_\mathcal{A}$ is not closed under contexts, $\gtrsim_\mathcal{A}$ is not a partial order, and $>_\mathcal{A}$ is not the strict part of $\gtrsim_\mathcal{A}$. Compatibility holds because of the identity $\gtrsim_\mathcal{A} \cdot >_\mathcal{A} = >_\mathcal{A}$.

In order for reduction pairs like $(\geq_{\text{lpo}}, >_{\text{lpo}})$ whose second component is closed under contexts to benefit from the fact that closure under contexts is not required, the conditions (1), (2), and (3) mentioned at the beginning of this section may be simplified by deleting certain (arguments of) function symbols occurring in $\mathcal{R}$ and $\mathcal{C}$ before testing orientability.

**Definition 16.** *An* argument filtering *for a signature $\mathcal{F}$ is a mapping $\pi$ that assigns to every $n$-ary function symbol $f \in \mathcal{F}$ an argument position $i \in \{1,\dots,n\}$ or a (possibly empty) list $[i_1,\dots,i_m]$ of argument positions with $1 \leqslant i_1 < \cdots < i_m \leqslant n$. The signature $\mathcal{F}_\pi$ consists of all function symbols $f$ such that $\pi(f)$ is some list $[i_1,\dots,i_m]$, where in $\mathcal{F}_\pi$ the arity of $f$ is $m$. Every argument filtering $\pi$ induces a mapping from $\mathcal{T}(\mathcal{F},\mathcal{V})$ to $\mathcal{T}(\mathcal{F}_\pi,\mathcal{V})$, also denoted by $\pi$:*

$$\pi(t) = \begin{cases} t & \text{if } t \text{ is a variable} \\ \pi(t_i) & \text{if } t = f(t_1,\dots,t_n) \text{ and } \pi(f) = i \\ f(\pi(t_{i_1}),\dots,\pi(t_{i_m})) & \text{if } t = f(t_1,\dots,t_n) \text{ and } \pi(f) = [i_1,\dots,i_m] \end{cases}$$

Note that the simple projections of the preceding sections can be viewed as special argument filterings.

*Example 17.* Applying the argument filtering $\pi$ with $\pi(\wedge) = \pi(\vee) = [\ ]$ and $\pi(\neg) = [1]$ to the rewrite rules of our leading example results in the following simplified rules:

$$\begin{aligned} 1: &\quad \neg\neg x \to x \\ 2: &\quad \neg(\vee) \to \wedge \\ 3: &\quad \neg(\wedge) \to \vee \\ 4: &\quad \wedge \to \vee \\ 5: &\quad \wedge \to \vee \end{aligned}$$

These rules are oriented from left to right by the lexicographic path order with precedence $\neg > \wedge > \vee$ (which does not imply termination of the original TRS.)

We are now ready to state and prove the standard dependency pair approach to the treatment of cycles in the dependency graph.

**Theorem 18 ([9]).** *Let $\mathcal{R}$ be a TRS and let $\mathcal{C}$ be a cycle in $\mathsf{DG}(\mathcal{R})$. If there exist an argument filtering $\pi$ and a reduction pair $(\gtrsim, >)$ such that $\pi(\mathcal{R}) \subseteq \gtrsim$, $\pi(\mathcal{C}) \subseteq \gtrsim \cup >$, and $\pi(\mathcal{C}) \cap > \neq \varnothing$ then there are no $\mathcal{C}$-minimal rewrite sequences.*

Although the condition $\pi(\mathcal{C}) \subseteq \gtrsim \cup >$ is weaker than $\pi(\mathcal{C}) \subseteq \gtrsim$, in practice there is no difference since all reduction pairs that are used in automatic tools satisfy the inclusion $> \subseteq \gtrsim$.

*Proof.* Suppose to the contrary that there exists a $\mathcal{C}$-minimal rewrite sequence:

$$t_1 \to^*_{\mathcal{R}} u_1 \to_{\mathcal{C}} t_2 \to^*_{\mathcal{R}} u_2 \to_{\mathcal{C}} t_3 \to^*_{\mathcal{R}} \cdots \tag{2}$$

We show that after applying the argument filtering $\pi$ we obtain an infinite descending sequence with respect to the well-founded order $>$. Let $i \geqslant 1$.

- First consider the dependency pair step $u_i \to_{\mathcal{C}} t_{i+1}$. Since $u_i \in \mathcal{T}^\sharp$, the step takes place at the root positions and thus there exist a dependency pair $l \to r \in \mathcal{C}$ and a substitution $\sigma$ such that $u_i = l\sigma$ and $t_{i+1} = r\sigma$. Define the substitution $\sigma_\pi$ as the composition of $\sigma$ and $\pi$, i.e., $\sigma_\pi(x) = \pi(\sigma(x))$ for every variable $x$. A straightforward induction proof reveals that $\pi(t\sigma) = \pi(t)\sigma_\pi$ for every term $t$. Hence $\pi(u_i) = \pi(l)\sigma_\pi$ and $\pi(t_{i+1}) = \pi(r)\sigma_\pi$. From the assumption $\pi(\mathcal{C}) \subseteq \gtrsim \cup >$ we infer that $\pi(l) \gtrsim \pi(r)$ or $\pi(l) > \pi(r)$. Since both $\gtrsim$ and $>$ are closed under substitutions, we have $\pi(u_i) \gtrsim \pi(t_{i+1})$ or $\pi(u_i) > \pi(t_{i+1})$. As in the proof of Theorem 11, the latter holds for infinitely many $i$ because of the assumption $\pi(\mathcal{C}) \cap > \neq \varnothing$.
- Next consider the rewrite sequence $t_i \to^*_{\mathcal{R}} u_i$. Using the assumption $\pi(\mathcal{R}) \subseteq \gtrsim$, we obtain $\pi(t_i) \gtrsim^* \pi(u_i)$ and thus $\pi(t_i) \gtrsim \pi(u_i)$ as in the preceding case.

So (2) is transformed into an infinite descending sequence consisting of $\gtrsim$ and $>$ steps, where there are an infinite number of the latter. Using the compatibility of $\gtrsim$ and $>$, we obtain an infinite descending sequence with respect to $>$, providing the desired contradiction. ☐

*Example 19.* The argument filtering of Example 17 cannot be used to handle the remaining SCC $\{11, 12, 13, 14\}$ in our leading example. This can be seen as follows. Because $\pi(\vee) = [\,]$, irrespective of the choice of $\pi(\wedge^\sharp)$, variables $y$ and $z$ will no longer appear in the left-hand sides of the simplified dependency pairs. Hence they cannot appear in the right-hand sides, and this is only possible if we take 1, [1], or [ ] for $\pi(\wedge^\sharp)$. The first two choices transform dependency pairs 13 and 14 into rules in which the variable $x$ appears on the right-hand side but not on the left-hand side, whereas the third choice turns all dependency pairs into the identity $\wedge^\sharp = \wedge^\sharp$.

Since the original TRS is compatible with the multiset path order, it is no surprise that the constraints of Theorem 18 for both SCCs are satisfied by the full argument filtering $\pi$ (that maps every $n$-ary function symbol to $[1, \ldots, n]$) and the reduction pair $(\geq_{\text{mpo}}, >_{\text{mpo}})$ with the precedence $\neg > \wedge > \vee$. However, it can be shown that there is no argument filtering $\pi$ such that the resulting constraints are satisfied by a polynomial interpretation or the lexicographic path order.

Observe that the proof of Theorem 18 does not use the fact that $\mathcal{C}$-minimal rewrite sequences start from terms in $\mathcal{T}^\sharp_\infty$. In the next section we show that by restoring the use of minimality, we can get rid of some of the constraints originating from $\mathcal{R}$.

## 5 Usable Rules

More precisely, we show that the concept of *usable* rules which was introduced in [1] to optimize the dependency pair method for *innermost* termination, can also be used for termination. The resulting termination criterion is stronger than previous results in this area ([10, 23]). We start by recalling the definition of usable rules.

**Definition 20.** *We write $f \blacktriangleright g$ if there exists a rewrite rule $l \to r \in \mathcal{R}$ such that $f = \mathrm{root}(l)$ and $g$ is a defined function symbol in $\mathcal{F}\mathsf{un}(r)$. For a set $\mathcal{G}$ of defined function symbols we denote by $\mathcal{R}{\restriction}\mathcal{G}$ the set of rewrite rules $l \to r \in \mathcal{R}$ with $\mathrm{root}(l) \in \mathcal{G}$. The set $\mathcal{U}(t)$ of usable rules of a term $t$ is defined as $\mathcal{R}{\restriction}\{g \mid f \blacktriangleright^* g \text{ for some } f \in \mathcal{F}\mathsf{un}(t)\}$. Finally, if $\mathcal{C}$ is a set of dependency pairs then*

$$\mathcal{U}(\mathcal{C}) = \bigcup_{l \to r \in \mathcal{C}} \mathcal{U}(r)$$

*Example 21.* None of the dependency pairs that appear in an SCC in our leading example have defined symbols in their right-hand sides, so for both SCCs the set of usable rules is empty. The same is true for the TRSs of Examples 13 and 14.

The following definition is the key to our result. It is a variation of a similar definition in Urbain [23], which in turn is based on a definition of Gramlich [13].

**Definition 22.** *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and let $\mathcal{G} \subseteq \mathcal{F}$. The* interpretation *$I_\mathcal{G}$ is a mapping from terminating terms in $\mathcal{T}(\mathcal{F}^\sharp, \mathcal{V})$ to terms in $\mathcal{T}(\mathcal{F}^\sharp \cup \{\mathsf{nil}, \mathsf{cons}\}, \mathcal{V})$, where $\mathsf{nil}$ and $\mathsf{cons}$ are fresh function symbols, inductively defined as follows:*

$$I_\mathcal{G}(t) = \begin{cases} t & \text{if } t \text{ is a variable} \\ f(I_\mathcal{G}(t_1), \dots, I_\mathcal{G}(t_n)) & \text{if } t = f(t_1, \dots, t_n) \text{ and } f \notin \mathcal{G} \\ \mathsf{cons}(f(I_\mathcal{G}(t_1), \dots, I_\mathcal{G}(t_n)), t') & \text{if } t = f(t_1, \dots, t_n) \text{ and } f \in \mathcal{G} \end{cases}$$

*where in the last clause $t'$ denotes the term $\mathrm{order}(\{I_\mathcal{G}(u) \mid t \to_\mathcal{R} u\})$ with*

$$\mathrm{order}(T) = \begin{cases} \mathsf{nil} & \text{if } T = \varnothing \\ \mathsf{cons}(t, \mathrm{order}(T \setminus \{t\})) & \text{if } t \text{ is the minimum element of } T \end{cases}$$

*Here we assume an arbitrary but fixed total order on $\mathcal{T}(\mathcal{F}^\sharp \cup \{\mathsf{nil}, \mathsf{cons}\}, \mathcal{V})$.*

Because we deal with finite TRSs, the relation is $\to_\mathcal{R}$ is finitely branching and hence the set $\{u \mid t \to_\mathcal{R} u\}$ of one-step reducts of $t$ is finite. Moreover, every term in this set is terminating. The well-definedness of $I_\mathcal{G}$ now follows by a straightforward induction argument. The difference with Urbain's definition is that we insert $f(I_\mathcal{G}(t_1), \dots, I_\mathcal{G}(t_n))$ in the list $t'$ when $f \in \mathcal{G}$. This modification is crucial for obtaining Theorem 29 below.

In the following $\mathcal{C}_{\mathcal{E}}$ denotes the TRS consisting of the two projection rules

$$\begin{aligned}\mathsf{cons}(x, y) &\to x\\ \mathsf{cons}(x, y) &\to y\end{aligned}$$

These rules are used to extract elements from the lists constructed by the interpretation $I_{\mathcal{G}}$. To improve readability, we abbreviate $\mathsf{cons}(t_1, \dots \mathsf{cons}(t_n, \mathsf{nil}) \dots)$ to $[t_1, \dots, t_n]$ in the next example.

*Example 23.* Consider the non-terminating TRS $\mathcal{R}$ consisting of the following three rewrite rules:

$$\begin{aligned}1\colon&\quad x + \mathsf{0} \to \mathsf{0}\\ 2\colon&\quad x \times \mathsf{0} \to \mathsf{0}\\ 3\colon&\quad x \times \mathsf{s}(y) \to (x + \mathsf{0}) \times \mathsf{s}(y)\end{aligned}$$

There are two dependency pairs:

$$\begin{aligned}4\colon&\quad x \times^\sharp \mathsf{s}(y) \to (x + \mathsf{0}) \times^\sharp \mathsf{s}(y)\\ 5\colon&\quad x \times^\sharp \mathsf{s}(y) \to x +^\sharp \mathsf{0}\end{aligned}$$

The dependency graph

$$\begin{tikzcd} 4 \arrow[loop above] \arrow[r] & 5 \end{tikzcd}$$

contains 1 cycle: $\mathcal{C} = \{4\}$. The following is a $\mathcal{C}$-minimal rewrite sequence:

$$\begin{aligned}((\mathsf{0}+\mathsf{0}) \times \mathsf{0}) \times^\sharp \mathsf{s}(\mathsf{0}) &\to_{\mathcal{C}} (((\mathsf{0}+\mathsf{0}) \times \mathsf{0}) + \mathsf{0}) \times^\sharp \mathsf{s}(\mathsf{0})\\ &\to_{\mathcal{R}} ((\mathsf{0}+\mathsf{0}) \times \mathsf{0}) \times^\sharp \mathsf{s}(\mathsf{0})\\ &\to_{\mathcal{R}} \mathsf{0} \times^\sharp \mathsf{s}(\mathsf{0})\\ &\to_{\mathcal{C}} (\mathsf{0}+\mathsf{0}) \times^\sharp \mathsf{s}(\mathsf{0})\\ &\to_{\mathcal{R}} \mathsf{0} \times^\sharp \mathsf{s}(\mathsf{0})\\ &\to_{\mathcal{C}} \cdots\end{aligned}$$

We have $\mathcal{U}(\mathcal{C}) = \{1\}$. Let $\mathcal{G}$ be the set of defined symbols of $\mathcal{R} \setminus \mathcal{U}(\mathcal{C})$, i.e., $\mathcal{G} = \{\times\}$. Applying the definition of $I_{\mathcal{G}}$ yields

$$\begin{aligned}I_{\mathcal{G}}(\mathsf{0} \times \mathsf{0}) &= \mathsf{cons}(I_{\mathcal{G}}(\mathsf{0}) \times I_{\mathcal{G}}(\mathsf{0}), \mathrm{order}(\{I_{\mathcal{G}}(\mathsf{0})\}))\\ &= \mathsf{cons}(\mathsf{0} \times \mathsf{0}, \mathrm{order}(\{\mathsf{0}\}))\\ &= \mathsf{cons}(\mathsf{0} \times \mathsf{0}, \mathsf{cons}(\mathsf{0}, \mathsf{nil}))\\ &= [\mathsf{0} \times \mathsf{0}, \mathsf{0}]\end{aligned}$$

and

$$\begin{aligned}I_{\mathcal{G}}((\mathsf{0}+\mathsf{0}) \times \mathsf{0}) &= \mathsf{cons}(I_{\mathcal{G}}(\mathsf{0}+\mathsf{0}) \times I_{\mathcal{G}}(\mathsf{0}), \mathrm{order}(\{I_{\mathcal{G}}(\mathsf{0} \times \mathsf{0}), I_{\mathcal{G}}(\mathsf{0})\}))\\ &= \mathsf{cons}((\mathsf{0}+\mathsf{0}) \times \mathsf{0}, \mathrm{order}(\{[\mathsf{0} \times \mathsf{0}, \mathsf{0}], \mathsf{0}\}))\\ &= [(\mathsf{0}+\mathsf{0}) \times \mathsf{0}, \mathsf{0}, [\mathsf{0} \times \mathsf{0}, \mathsf{0}]]\end{aligned}$$

if we assume that $\mathsf{0}$ is smaller than $[\mathsf{0} \times \mathsf{0}, \mathsf{0}]$ in the given total order. Now, by applying $I_{\mathcal{G}}$ to all terms in the above $\mathcal{C}$-minimal rewrite sequence, we obtain the following infinite rewrite sequence in $\mathcal{U}(\mathcal{C}) \cup \mathcal{C} \cup \mathcal{C}_{\mathcal{E}}$:

$$\begin{aligned} [(\mathsf{0}+\mathsf{0}) \times \mathsf{0}, \mathsf{0}, [\mathsf{0} \times \mathsf{0}, \mathsf{0}]] \times^{\sharp} \mathsf{s}(\mathsf{0}) &\to_{\mathcal{C}} ([(\mathsf{0}+\mathsf{0}) \times \mathsf{0}, \mathsf{0}, [\mathsf{0} \times \mathsf{0}, \mathsf{0}]] + \mathsf{0}) \times^{\sharp} \mathsf{s}(\mathsf{0}) \\ &\to_{\mathcal{U}(\mathcal{C})} [(\mathsf{0}+\mathsf{0}) \times \mathsf{0}, \mathsf{0}, [\mathsf{0} \times \mathsf{0}, \mathsf{0}]] \times^{\sharp} \mathsf{s}(\mathsf{0}) \\ &\to^{+}_{\mathcal{C}_{\mathcal{E}}} \mathsf{0} \times^{\sharp} \mathsf{s}(\mathsf{0}) \\ &\to_{\mathcal{C}} (\mathsf{0}+\mathsf{0}) \times^{\sharp} \mathsf{s}(\mathsf{0}) \\ &\to_{\mathcal{U}(\mathcal{C})} \mathsf{0} \times^{\sharp} \mathsf{s}(\mathsf{0}) \\ &\to_{\mathcal{C}} \cdots \end{aligned}$$

We start with some preliminary results. The first one addresses the behaviour of $I_{\mathcal{G}}$ on instantiated terms. The second states that $I_{\mathcal{G}}$ preserves any top part without $\mathcal{G}$-symbols.

**Definition 24.** *If $\sigma$ is a substitution that assigns to every variable in its domain a terminating term then we denote the substitution that assigns to every variable $x$ the term $I_{\mathcal{G}}(\sigma(x))$ by $\sigma_{I_{\mathcal{G}}}$.*

**Lemma 25.** *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and let $\mathcal{G} \subseteq \mathcal{F}$. Let $t$ be a term and $\sigma$ a substitution. If $t\sigma$ is terminating then $I_{\mathcal{G}}(t\sigma) \to^{*}_{\mathcal{C}_{\mathcal{E}}} t\sigma_{I_{\mathcal{G}}}$ and, if $t$ does not contain $\mathcal{G}$-symbols, $I_{\mathcal{G}}(t\sigma) = t\sigma_{I_{\mathcal{G}}}$.*

*Proof.* We use induction on $t$. If $t$ is a variable then $I_{\mathcal{G}}(t\sigma) = I_{\mathcal{G}}(\sigma(t)) = t\sigma_{I_{\mathcal{G}}}$. Let $t = f(t_1, \ldots, t_n)$. We distinguish two cases.

1. If $f \notin \mathcal{G}$ then $I_{\mathcal{G}}(t\sigma) = f(I_{\mathcal{G}}(t_1\sigma), \ldots, I_{\mathcal{G}}(t_n\sigma))$. The induction hypothesis yields $I_{\mathcal{G}}(t_i\sigma) \to^{*}_{\mathcal{C}_{\mathcal{E}}} t_i\sigma_{I_{\mathcal{G}}}$ for $1 \leqslant i \leqslant n$ and thus
$$I_{\mathcal{G}}(t\sigma) \to^{*}_{\mathcal{C}_{\mathcal{E}}} f(t_1\sigma_{I_{\mathcal{G}}}, \ldots, t_n\sigma_{I_{\mathcal{G}}}) = t\sigma_{I_{\mathcal{G}}}$$
If there are no $\mathcal{G}$-symbols in $t_1, \ldots, t_n$ then we obtain $I_{\mathcal{G}}(t_i\sigma) = t_i\sigma_{I_{\mathcal{G}}}$ for all $1 \leqslant i \leqslant n$ from the induction hypothesis and thus $I_{\mathcal{G}}(t\sigma) = t\sigma_{I_{\mathcal{G}}}$.
2. If $f \in \mathcal{G}$ then
$$I_{\mathcal{G}}(t\sigma) = \mathsf{cons}(f(I_{\mathcal{G}}(t_1\sigma), \ldots, I_{\mathcal{G}}(t_n\sigma)), t') \to_{\mathcal{C}_{\mathcal{E}}} f(I_{\mathcal{G}}(t_1\sigma), \ldots, I_{\mathcal{G}}(t_n\sigma))$$
for some term $t'$. We obtain $f(I_{\mathcal{G}}(t_1\sigma), \ldots, I_{\mathcal{G}}(t_n\sigma)) \to^{*}_{\mathcal{C}_{\mathcal{E}}} t\sigma_{I_{\mathcal{G}}}$ as in the preceding case and thus $I_{\mathcal{G}}(t\sigma) \to^{*}_{\mathcal{C}_{\mathcal{E}}} t\sigma_{I_{\mathcal{G}}}$ as desired.

□

The preceding lemma is not true for Urbain's interpretation function.

**Lemma 26.** *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and let $\mathcal{G} \subseteq \mathcal{F}$. If $t = C[t_1, \ldots, t_n]$ is terminating and the context $C$ contains no $\mathcal{G}$-symbols then $I_{\mathcal{G}}(t) = C[I_{\mathcal{G}}(t_1), \ldots, I_{\mathcal{G}}(t_n)]$.*

*Proof.* Let $t'$ be the term $C[x_1, \dots, x_n]$ where $x_1, \dots, x_n$ are fresh variables. We have $t = t'\sigma$ for the substitution $\sigma = \{x_i \mapsto t_i \mid 1 \leqslant i \leqslant n\}$. The preceding lemma yields $I_\mathcal{G}(t) = t'\sigma_{I_\mathcal{G}}$. Clearly $t'\sigma_{I_\mathcal{G}} = C[I_\mathcal{G}(t_1), \dots, I_\mathcal{G}(t_n)]$. □

The next lemma states an easy connection between usable rules and defined symbols of the other rules.

**Lemma 27.** *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and let $\mathcal{C} \subseteq \mathsf{DP}(\mathcal{R})$. Furthermore, let $\mathcal{G}$ be the set of defined symbols of $\mathcal{R} \setminus \mathcal{U}(\mathcal{C})$.*

1. *$\mathcal{R} = \mathcal{U}(\mathcal{C}) \cup (\mathcal{R}{\restriction}\mathcal{G})$.*
2. *If $l \to r \in \mathcal{U}(\mathcal{C})$ then $r$ contains no $\mathcal{G}$-symbols.*

*Proof.* The first statement is obvious. For the second statement we reason as follows. Suppose to the contrary that $r$ contains a function symbol $g \in \mathcal{G}$. We have $l \to r \in \mathcal{U}(t)$ for some $s \to t \in \mathcal{C}$. So there exists a function symbol $f \in \mathcal{F}\mathsf{un}(t)$ such that $f \blacktriangleright^* \mathrm{root}(l)$. We have $\mathrm{root}(l) \blacktriangleright g$ by the definition of $\blacktriangleright$ and hence also $f \blacktriangleright^* g$. Therefore $\mathcal{R}{\restriction}\{g\} \subseteq \mathcal{U}(t) \subseteq \mathcal{U}(\mathcal{C})$. So $g$ is a defined symbol of a rule in $\mathcal{U}(\mathcal{C})$. This contradicts the assumption that $g \in \mathcal{G}$. □

The following lemma is the key result for the new termination criterion. It states that rewrite steps in $\mathcal{R}$ are transformed by $I_\mathcal{G}$ into rewrite sequences in $\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_\mathcal{E}$, provided $\mathcal{G}$ is the set of defined symbols of $\mathcal{R} \setminus \mathcal{U}(\mathcal{C})$.

**Lemma 28.** *Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and let $\mathcal{C} \subseteq \mathsf{DP}(\mathcal{R})$. Furthermore, let $\mathcal{G}$ be the set of defined symbols of $\mathcal{R} \setminus \mathcal{U}(\mathcal{C})$. If terms $s$ and $t$ are terminating and $s \to_\mathcal{R} t$ then $I_\mathcal{G}(s) \to^+_{\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_\mathcal{E}} I_\mathcal{G}(t)$.*

*Proof.* Let $p$ be the position of the rewrite step $s \to_\mathcal{R} t$. We distinguish two cases.

- First suppose that there is a function symbol from $\mathcal{G}$ at a position $q \leqslant p$. In this case we may write $s = C[s_1, \dots, s_i, \dots, s_n]$ and $t = C[s_1, \dots, t_i, \dots, s_n]$ with $s_i \to_\mathcal{R} t_i$, where $\mathrm{root}(s_i) \in \mathcal{G}$ and the context $C$ contains no $\mathcal{G}$-symbols. We have $I_\mathcal{G}(s_i) \to_{\mathcal{C}_\mathcal{E}} \mathrm{order}(\{I_\mathcal{G}(u) \mid s_i \to_\mathcal{R} u\})$. Since $s_i \to_\mathcal{R} t_i$, we can extract $I_\mathcal{G}(t_i)$ from the term $\mathrm{order}(\{I_\mathcal{G}(u) \mid s_i \to_\mathcal{R} u\})$ by appropriate $\mathcal{C}_\mathcal{E}$ steps, so $I_\mathcal{G}(s_i) \to^+_{\mathcal{C}_\mathcal{E}} I_\mathcal{G}(t_i)$. We now obtain $I_\mathcal{G}(s) \to^+_{\mathcal{C}_\mathcal{E}} I_\mathcal{G}(t)$ from Lemma 26.
- In the other case $s = C[s_1, \dots, s_i, \dots, s_n]$ and $t = C[s_1, \dots, t_i, \dots, s_n]$ with $s_i \xrightarrow{\epsilon}_\mathcal{R} t_i$, where $\mathrm{root}(s_i) \notin \mathcal{G}$ and the context $C$ contains no $\mathcal{G}$-symbols. Since $\mathrm{root}(s_i) \notin \mathcal{G}$ the applied rewrite rule $l \to r$ in the step $s_i \xrightarrow{\epsilon}_\mathcal{R} t_i$ must come from $\mathcal{U}(\mathcal{C})$ according to part 1 of Lemma 27. Let $\sigma$ be the substitution with $\mathcal{D}\mathsf{om}(\sigma) \subseteq \mathcal{V}\mathsf{ar}(l)$ such that $s_i = l\sigma$ and $t_i = r\sigma$. According to part 2 of Lemma 27, $r$ contains no $\mathcal{G}$-symbols and thus we obtain $I_\mathcal{G}(s_i) \to^*_{\mathcal{C}_\mathcal{E}} l\sigma_{I_\mathcal{G}}$ and $I_\mathcal{G}(t_i) = r\sigma_{I_\mathcal{G}}$ from Lemma 25. Clearly $l\sigma_{I_\mathcal{G}} \to_{\mathcal{U}(\mathcal{C})} r\sigma_{I_\mathcal{G}}$ and thus $I_\mathcal{G}(s_i) \to^+_{\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_\mathcal{E}} I_\mathcal{G}(t_i)$. Lemma 26 now yields the desired $I_\mathcal{G}(s) \to^+_{\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_\mathcal{E}} I_\mathcal{G}(t)$.

□

After these preparations, the main result[2] of this section is now easily proved.

**Theorem 29.** *Let $\mathcal{R}$ be a TRS and let $\mathcal{C}$ be a cycle in $\mathsf{DG}(\mathcal{R})$. If there exist an argument filtering $\pi$ and a reduction pair $(\gtrsim, >)$ such that $\pi(\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_{\mathcal{E}}) \subseteq \gtrsim$, $\pi(\mathcal{C}) \subseteq \gtrsim \cup >$, and $\pi(\mathcal{C}) \cap > \neq \varnothing$ then there are no $\mathcal{C}$-minimal rewrite sequences.*

*Proof.* Suppose to the contrary that there exists a $\mathcal{C}$-minimal rewrite sequence:

$$t_1 \to^*_{\mathcal{R}} u_1 \to_{\mathcal{C}} t_2 \to^*_{\mathcal{R}} u_2 \to_{\mathcal{C}} t_3 \to^*_{\mathcal{R}} \cdots \tag{3}$$

Let $\mathcal{G}$ be the set of defined symbols of $\mathcal{R} \setminus \mathcal{U}(\mathcal{C})$. We show that after applying the interpretation $I_{\mathcal{G}}$ we obtain an infinite rewrite sequence in $\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_{\mathcal{E}} \cup \mathcal{C}$ in which every rule of $\mathcal{C}$ is used infinitely often. Since all terms in (3) belong to $\mathcal{T}^{\sharp}_{\infty}$, they are terminating with respect to $\mathcal{R}$ and hence we can indeed apply the interpretation $I_{\mathcal{G}}$. Let $i \geqslant 1$.

- First consider the dependency pair step $u_i \to_{\mathcal{C}} t_{i+1}$. There exist a dependency pair $l \to r \in \mathcal{C}$ and a substitution $\sigma$ such that $u_i = l\sigma$ and $t_{i+1} = r\sigma$. We may assume that $\mathcal{D}\mathsf{om}(\sigma) \subseteq \mathcal{V}\mathsf{ar}(l)$. Since $u_i \in \mathcal{T}^{\sharp}_{\infty}$, $\sigma(x)$ is terminating for every variable $x \in \mathcal{V}\mathsf{ar}(l)$. Hence the substitution $\sigma_{I_{\mathcal{G}}}$ is well-defined. Since $r$ lacks $\mathcal{G}$-symbols by Lemma 27, we have $I_{\mathcal{G}}(r\sigma) = r\sigma_{I_{\mathcal{G}}}$ by Lemma 25. Furthermore, $I_{\mathcal{G}}(l\sigma) \to^*_{\mathcal{C}_{\mathcal{E}}} l\sigma_{I_{\mathcal{G}}}$ by Lemma 25. Hence
$$I_{\mathcal{G}}(u_i) \to^*_{\mathcal{C}_{\mathcal{E}}} l\sigma_{I_{\mathcal{G}}} \to_{\mathcal{C}} r\sigma_{I_{\mathcal{G}}} = I_{\mathcal{G}}(t_{i+1})$$
- Next consider the rewrite sequence $t_i \to^*_{\mathcal{R}} u_i$. Because all terms in this sequence are terminating, we obtain $I_{\mathcal{G}}(t_i) \to^*_{\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_{\mathcal{E}}} I_{\mathcal{G}}(u_i)$ by repeated applications of Lemma 28.

Next we apply the argument filtering $\pi$ to all terms in the resulting infinite rewrite sequence in $\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_{\mathcal{E}} \cup \mathcal{C}$. Because of the assumptions of this theorem, we can simply reuse the proof of Theorem 18 (where $\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_{\mathcal{E}}$ takes the place of $\mathcal{R}$) and obtain the desired contradiction with the well-foundedness of $>$. □

Since $\mathcal{U}(\mathcal{C})$ in general is a proper subset of $\mathcal{R}$, the condition $\pi(\mathcal{U}(\mathcal{C})) \subseteq \gtrsim$ is easier to satisfy than the condition $\pi(\mathcal{R}) \subseteq \gtrsim$ of Theorem 18. What about the additional condition $\pi(\mathcal{C}_{\mathcal{E}}) \subseteq \gtrsim$? By choosing $\pi(\mathsf{cons}) = [1, 2]$ the condition reduces to $\mathsf{cons}(x, y) \gtrsim x$ and $\mathsf{cons}(x, y) \gtrsim y$. Virtually all reduction pairs that are used in termination tools can be extended to satisfy this condition. For reduction pairs that are based on simplification orders, like $(\geq_{\mathrm{lpo}}, >_{\mathrm{lpo}})$, this is clear. A sufficient condition that makes the semantic construction described in Section 4 for generating reduction pairs work is that each pair of elements of the carrier has a least upper bound. For interpretations in the set $\mathbb{N}$ of natural numbers equipped with the standard order this is obviously satisfied. The necessity of the least upper bound condition follows by considering the term algebra associated with the famous rule $\mathsf{f}(\mathsf{a}, \mathsf{b}, x) \to \mathsf{f}(x, x, x)$ of Toyama [22] equipped with the well-founded order $\to^+$.

[2] This result has been independently obtained by Thiemann *et al.* [21].

As a matter of fact, due to the condition $\pi(\mathcal{C}_\mathcal{E}) \subseteq \gtrsim$, Theorem 29 provides only a sufficient condition for the absence of $\mathcal{C}$-minimal rewrite sequences. A concrete example of a terminating TRS that cannot be proved terminating by the criterion of Theorem 29 will be presented at the end of this section.

*Example 30.* Let us take a final look at the SCC $\{11, 12, 13, 14\}$ in our leading example. There are no usable rules. By taking the linear polynomial interpretation $\wedge^\sharp{}_\mathbb{N}(x, y) = x + y$ and $\vee_\mathbb{N}(x, y) = x + y + 1$ the involved dependency pairs reduce the following inequalities:

$$\begin{aligned} 11\colon &\quad x + y + z + 1 > x + y \\ 12\colon &\quad x + y + z + 1 > x + z \\ 13\colon &\quad x + y + z + 1 > x + y \\ 14\colon &\quad x + y + z + 1 > x + z \end{aligned}$$

Hence there are no $\mathcal{C}$-minimal rewrite sequences for any nonempty subset $\mathcal{C} \subseteq \{11, 12, 13, 14\}$ and we conclude that the TRS is terminating.

The modularity result in Giesl *et al.* [10] can be expressed as the version of Theorem 29 where $\mathcal{U}(\mathcal{C})$ is replaced by

$$\mathcal{U}'(\mathcal{C}) = \bigcup_{l \to r \in \mathcal{C}} \mathcal{U}'(l^\flat)$$

The mapping $(\cdot)^\flat \colon \mathcal{T}^\sharp \to \mathcal{T}$ replaces the dependency pair symbol $f^\sharp$ at the root of its argument by the original defined function symbol $f$ and $\mathcal{U}'(t)$ is computed like $\mathcal{U}(t)$ but with a different relation $\blacktriangleright'$ that relates more function symbols: $f \blacktriangleright' g$ if there exists a rewrite rule $l \to r \in \mathcal{R}$ such that $f = \mathrm{root}(l)$ and $g$ is a defined function symbol in $\mathcal{F}\mathsf{un}(l) \cup \mathcal{F}\mathsf{un}(r)$.

Since $\mathcal{U}(r) \subseteq \mathcal{U}(r^\flat) \subseteq \mathcal{U}(l^\flat) \subseteq \mathcal{U}'(l^\flat)$ for every dependency pair $l \to r$, it is clear that $\mathcal{U}(\mathcal{C})$ is always a subset of $\mathcal{U}'(\mathcal{C})$. Very often it is a proper subset and that may affect the ability to prove termination. This will become clear from the experimental data in the next section.

*Example 31.* If we adopt the above definition of usable rules then for the SCC $\{7, 8, 9, 10\}$ in our leading example all five rewrite rules are usable whereas for the SCC $\{11, 12, 13, 14\}$ only rules 4 and 5 are usable. For the SCC $\{9\}$ in Example 13 all seven rewrite rules and for the SCC $\{10, 11, 12\}$ rules 3–7 are usable. Finally, for the SCC $\{7\}$ in Example 14 rules 1 and 2 are usable whereas for the SCC $\{8, 10\}$ all six rewrite rules are usable.

Combining the two main results of this paper, we arrive at the following corollary.

**Corollary 32.** *A TRS $\mathcal{R}$ is terminating if for every cycle $\mathcal{C}$ in $\mathsf{DG}(\mathcal{R})$ one of the following two conditions holds:*

– *there exists a simple projection $\pi$ for $\mathcal{C}$ such that $\pi(\mathcal{C}) \subseteq \trianglerighteq$ and $\pi(\mathcal{C}) \cap \triangleright \neq \varnothing$,*

– *there exist an argument filtering $\pi$ and a reduction pair $(\gtrsim, >)$ such that $\pi(\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_{\mathcal{E}}) \subseteq \gtrsim$, $\pi(\mathcal{C}) \subseteq \gtrsim \cup >$, and $\pi(\mathcal{C}) \cap > \neq \varnothing$.*

□

The final example in this paper shows that the reverse does not hold. This is in contrast to Theorem 18, which provides a sufficient and necessary condition for termination. The reason is that termination of a TRS $\mathcal{R}$ is equivalent to the termination of $\mathcal{R} \cup \mathsf{DP}(\mathcal{R})$, a result due to [1] (see [18] for a simple proof based on type introduction).

*Example 33.* Consider the terminating TRS $\mathcal{R}$ consisting of the following two rewrite rules:

$$\begin{aligned} 1\colon \quad \mathsf{f}(\mathsf{s}(\mathsf{a}), \mathsf{s}(\mathsf{b}), x) &\to \mathsf{f}(x, x, x) \\ 2\colon \quad \mathsf{g}(\mathsf{f}(\mathsf{s}(x), \mathsf{s}(y), z)) &\to \mathsf{g}(\mathsf{f}(x, y, z)) \end{aligned}$$

There are three dependency pairs:

$$\begin{aligned} 3\colon \quad \mathsf{f}^\sharp(\mathsf{s}(\mathsf{a}), \mathsf{s}(\mathsf{b}), x) &\to \mathsf{f}^\sharp(x, x, x) \\ 4\colon \quad \mathsf{g}^\sharp(\mathsf{f}(\mathsf{s}(x), \mathsf{s}(y), z)) &\to \mathsf{g}^\sharp(\mathsf{f}(x, y, z)) \\ 5\colon \quad \mathsf{g}^\sharp(\mathsf{f}(\mathsf{s}(x), \mathsf{s}(y), z)) &\to \mathsf{f}^\sharp(x, y, z) \end{aligned}$$

The dependency graph

$$\begin{tikzcd} 3 & 5 \arrow[l] & 4 \arrow[l] \arrow[loop above] \end{tikzcd}$$

contains 1 cycle: $\mathcal{C} = \{4\}$. The only simple projection for $\mathsf{g}^\sharp$ transforms 4 into

$$4\colon \quad \mathsf{f}(\mathsf{s}(x), \mathsf{s}(y), z) \to \mathsf{f}(x, y, z)$$

and $\mathsf{f}(x, y, z)$ is not a proper subterm of $\mathsf{f}(\mathsf{s}(x), \mathsf{s}(y), z)$. We have $\mathcal{U}(\mathcal{C}) = \{1\}$. We claim that the inclusions $\pi(\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_{\mathcal{E}}) \subseteq \gtrsim$ and $\pi(\mathcal{C}) \subseteq >$ are not satisfied for any argument filtering $\pi$ and reduction pair $(\gtrsim, >)$. The reason is simply that the term $t = \mathsf{g}^\sharp(\mathsf{f}(u, u, u))$ with $u = \mathsf{s}(\mathsf{cons}(\mathsf{s}(\mathsf{a}), \mathsf{s}(\mathsf{b})))$ admits the following cyclic reduction in $\mathcal{U}(\mathcal{C}) \cup \mathcal{C}_{\mathcal{E}} \cup \mathcal{C}$:

$$\begin{aligned} t &\to_{\mathcal{C}} \mathsf{g}^\sharp(\mathsf{f}(\mathsf{cons}(\mathsf{s}(\mathsf{a}), \mathsf{s}(\mathsf{b})), \mathsf{cons}(\mathsf{s}(\mathsf{a}), \mathsf{s}(\mathsf{b})), u)) \\ &\to_{\mathcal{C}_{\mathcal{E}}} \mathsf{g}^\sharp(\mathsf{f}(\mathsf{s}(\mathsf{a}), \mathsf{cons}(\mathsf{s}(\mathsf{a}), \mathsf{s}(\mathsf{b})), u)) \\ &\to_{\mathcal{C}_{\mathcal{E}}} \mathsf{g}^\sharp(\mathsf{f}(\mathsf{s}(\mathsf{a}), \mathsf{s}(\mathsf{b}), u)) \\ &\to_{\mathcal{U}(\mathcal{C})} t \end{aligned}$$

## 6 Benchmarks

We implemented the new criteria presented in the preceding sections in the Tyrolean Termination Tool [16], the successor of the Tsukuba Termination Tool [15].

We tested the effect of the improvements described in the previous sections on 223 examples from three different sources:

– all 89 terminating TRSs from Arts and Giesl [2],
– all 23 TRSs from Dershowitz [7],
– all 119 terminating TRSs from Steinbach and Kühler [19, Sections 3 and 4].

Eight of these TRSs appear in more than one collection, so the total number is 223. In all experiments we used the $\mathsf{EDG}^*$ approximation [17] of the dependency graph and, when the lexicographic path order is used, the divide and conquer algorithm described in the full version of [14] is used to search for suitable argument filterings. The experiments were performed on a PC equipped with a 2.20 GHz Mobile Intel Pentium 4 Processor - M and 512 MB of memory.

The results are summarized in Table 1. The letters in the column headings have the following meaning:

| | |
|---|---|
| `s` | the subterm criterion of Section 3, |
| `u` | the usable rules criterion of Section 5, |
| `l` | lexicographic path order in combination with the argument filtering heuristic that considers for an $n$-ary function symbol the full argument filtering $[1,\ldots,n]$ in addition to the $n$ collapsing argument filterings $1, \ldots, n$, |
| `p` | polynomial interpretation restricted to linear polynomials with coefficients from $\{0,1\}$; the usefulness of the latter restriction has been first observed in [11]. |

We list the number of successful termination attempts, the number of failures (which means that no termination proof was found while fully exploring the search space implied by the options), and the number of timeouts, which we set to 30 seconds. The numbers in parentheses refer to the usable rules criterion of [10] which is described in the latter part of Section 5. The figures below the number of successes and failures indicate the average time in seconds. It is interesting to note that the subterm criterion could handle 279 of the 395 generated SCCs, resulting in termination proofs for 128 of the 223 TRSs.

## References

1. T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000.
2. T. Arts and J. Giesl. A collection of examples for termination of term rewriting using dependency pairs. Technical Report AIB-2001-09, RWTH Aachen, 2001.
3. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
4. L. Bachmair and N. Dershowitz. Commutation, transformation, and termination. In *Proceedings of the 8th International Conference on Automated Deduction*, volume 230 of *Lecture Notes in Computer Science*, pages 5–20, 1986.
5. C. Borralleras, M. Ferreira, and A. Rubio. Complete monotonic semantic path orderings. In *Proceedings of the 17th International Conference on Automated Deduction*, volume 1831 of *Lecture Notes in Artificial Intelligence*, pages 346–364, 2000.
6. E. Contejean, C. Marché, B. Monate, and X. Urbain. C*i*ME version 2, 2000. Available at `http://cime.lri.fr/`.

**Table 1.** Summary.

| | `s` | `l` | `sl` | `ul` | `sul` |
|---|---|---|---|---|---|
| success | 128 | 133 | 149 | 144 (138) | 152 (151) |
| | 0.01 | 0.07 | 0.01 | 0.01 (0.02) | 0.01 (0.01) |
| failure | 95 | 90 | 74 | 79 (85) | 71 (72) |
| | 0.01 | 0.01 | 0.01 | 0.01 (0.02) | 0.01 (0.02) |
| timeout | 0 | 0 | 0 | 0 (0) | 0 (0) |
| total time | 1.72 | 10.49 | 2.33 | 2.31 (4.46) | 2.07 (2.80) |
| | | `p` | `sp` | `up` | `sup` |
| success | | 139 | 180 | 179 (148) | 189 (185) |
| | | 0.32 | 0.37 | 0.28 (0.33) | 0.25 (0.40) |
| failure | | 77 | 39 | 44 (71) | 34 (37) |
| | | 0.52 | 0.33 | 0.03 (0.66) | 0.03 (0.59) |
| timeout | | 7 | 4 | 0 (4) | 0 (1) |
| total time | | 294.13 | 198.83 | 51.30 (215.93) | 47.79 (125.97) |

7. N. Dershowitz. 33 Examples of termination. In *French Spring School of Theoretical Computer Science*, volume 909 of *Lecture Notes in Computer Science*, pages 16–26, 1995.
8. N. Dershowitz. Termination dependencies. In *Proceedings of the 6th International Workshop on Termination*, Technical Report DSIC-II/15/03, Universidad Politécnica de Valencia, pages 27–30, 2003.
9. J. Giesl, T. Arts, and E. Ohlebusch. Modular termination proofs for rewriting using dependency pairs. *Journal of Symbolic Computation*, 34(1):21–58, 2002.
10. J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Improving dependency pairs. In *Proceedings of the 10th International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 2850 of *Lecture Notes in Artificial Intelligence*, pages 165–179, 2003.
11. J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing dependency pairs. Technical Report AIB-2003-08, RWTH Aachen, Germany, 2003.
12. J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Automated termination proofs with AProVE. In *Proceedings of the 15th International Conference on Rewriting Techniques and Applications*, Lecture Notes in Computer Science, 2004. This volume.
13. B. Gramlich. Generalized sufficient conditions for modular termination of rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 5:131–158, 1994.
14. N. Hirokawa and A. Middeldorp. Automating the dependency pair method. In *Proceedings of the 19th International Conference on Automated Deduction*, volume 2741 of *Lecture Notes in Artificial Intelligence*, pages 32–46, 2003. Full version submitted for publication.

15. N. Hirokawa and A. Middeldorp. Tsukuba termination tool. In *Proceedings of the 14th International Conference on Rewriting Techniques and Applications*, volume 2706 of *Lecture Notes in Computer Science*, pages 311–320, 2003.
16. N. Hirokawa and A. Middeldorp. Tyrolean termination tool, 2004. Available at `http://cl2-informatik.uibk.ac.at/ttt`.
17. A. Middeldorp. Approximations for strategies and termination. In *Proceedings of the 2nd International Workshop on Reduction Strategies in Rewriting and Programming*, volume 70(6) of *Electronic Notes in Theoretical Computer Science*, 2002.
18. A. Middeldorp and H. Ohsaki. Type introduction for equational rewriting. *Acta Informatica*, 36(12):1007–1029, 2000.
19. J. Steinbach and U. Kühler. Check your ordering – termination proofs and open problems. Technical Report SR-90-25, Universität Kaiserslautern, 1990.
20. Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.
21. R. Thiemann, J. Giesl, and P. Schneider-Kamp. Improved modular termination proofs using dependency pairs. In *Proceedings of the 2nd International Joint Conference on Automated Reasoning*, Lecture Notes in Artificial Intelligence, 2004. To appear.
22. Y. Toyama. Counterexamples to the termination for the direct sum of term rewriting systems. *Information Processing Letters*, 25:141–143, 1987.
23. X. Urbain. Modular & incremental automated termination proofs. *Journal of Automated Reasoning*, 2004. To appear.
24. H. Zantema. TORPA: Termination of rewriting proved automatically. In *Proceedings of the 15th International Conference on Rewriting Techniques and Applications*, Lecture Notes in Computer Science, 2004. This volume.